- Safe Vault

[Draft]

September 2023

Abstract

Safe Vault introduces a set of libraries called “plugins” in the new Safe protocol, which can enhance and enforce security for individual Safe wallets, as well as multi-sigs, it would do so by introducing different plugins onto the new safe{core} by making use of Hooks and other types of plugins, we can build tools such as, denyList, whiteList, but also Access role-based plugins which would need certain guardRails around what they should and should not do.

Introduction

The smart contract ecosystem has experienced a tremendous growth in recent years, coupled with a surge in adoption of web3 wallets, however that expansion has brought forth a new set of challenges, ranging from unexpected obstacles to basic issues.

Phishing attacks stand out as one of the most annoying, and continuously persistent, making use of simplicity of the ERC-20 standard, and the user’s inability to decode calldata, phishers have made up to 100+ million dollars and much more, simply by conning the legitimacy of their website to unbeknownst users.

In response to these issues, we introduce a system designed not to only handle these security problems, but any that could come to face a safe smart contract, extending on the modularity, we introduce guards that work together to protect users from malicious tx’s pre-execution, if detected that the transaction is sent to a blacklisted address (known phisher/malicious actors), we automatically assume that the user’s EOA has been compromised and swap owners with a new owner where the

user is assigned that new EOA using safes' recovery modules, such as social recovery or other self-defined options, which can be selected before enabling the module.

Expanding on this foundation, many different use cases emerge, where a safe would have an admin, who signs transactions regularly, and it they are to sign a transaction containing a "to" field either within the blacklist or outside of an allowlist, it is assumed that the user has turned rogue/lost their EOA, and are assigned back to regular user with their new EOA until further action by the multi-sig users.

Features

DenyList/AllowList

Denylist, a safe module that defines a set of functions (through their methodId) that are not permitted to be signed in the safe, as well as a sister contract which defines the only functions that can be signed, serving as a allowList contract. One or both can both can be enabled

Role-based admin-access

A typical role-based module, allowing a safe to define it's Admin, Semi-admin, as well as potential non-admins, typically not ideal, however coupled with the previous module, becomes well defined, giving non-admins specific duties to sign transactions they are expected to sign, and given admins the privilege to deny-transactions as well as having an additional layer of security with when integrated with deny/allow list modules

Admin-Social recovery module

Delivering on the Vault structure, we present a social recovery module, that when activated, the user provides their social account they deem they are capable of recovering, from there, once enabled, the user’s Safe is no longer connected to their EOA, since in the event of their private key’s leak, or malicious signature replay’s, we detect said transactions through the denyList, and once the tx is confirmed as malicious, we perform an Ownerswap with a new one sent to their trusted social account, where they can regain control over their safe by simple importing their new EOA